

Handling forms in PHP

Table of Contents

HTML forms.....	1
Summary of HTML form elements.....	1
Capturing form data with PHP.....	1
Handling empty form fields	4
Dealing with multi value fields.....	7
rtrim	9
Passing data between servers and browsers.....	9
Stateless	9
Error checking web forms	9

An important part of any web application is the ability to accept input from the user and interact with the user. One of the most common ways to capture data from the user is through HTML forms.



HTML forms

All web forms start with an opening `<form>` tag and end with a closing `</form>` tag.

```
<form action="myPage.php" method="post">
```

```
...
```

```
</form>
```

The action attribute tells the web browser where to send the form data to. Method tells the browser how to send the data. The method can be set to get (good for small amounts of data) and post (better for larger amounts of data).

Summary of HTML form elements

If you need a refresh have a look at: http://www.w3schools.com/html/html_forms.asp

Capturing form data with PHP

To read data from a form you need to use super global variables. A super global is a built in PHP variable that is available in any scope. We will be using 3 super global arrays:

Super global array	Description
<code>\$_GET</code>	Contains a list of all the field names and values sent by a form using the get method
<code>\$_POST</code>	Contains a list of all the field names and values sent using post
<code>\$_REQUEST</code>	Contains the values of both <code>\$_GET</code> and <code>\$_POST</code> along with the values of the <code>\$_COOKIE</code> super global array

Each of these three super global arrays contains the field names from the sent form as array keys, with the field values themselves as array values. For example, say you created a form using the get method, and that form contained the following control:

```
<input type="text" name="emailAddress" value="" >
```

You could then access the value that the user entered into that form field using either the `$_GET` or the `$_REQUEST` superglobal:

```
$email = $_GET["emailAddress"];
```

```
$email = $_REQUEST["emailAddress"];
```

Activity 1. Registration form

For this activity download the registration form and stylesheet.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Registration form</title>
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <form action="processRegistration.php" method="post">
    <fieldset>
      <h1>Registration Form</h1>
      <p>Please fill out the details to register</p>
      <div>
        <label for="firstName">First name</label>
        <input type="text" name="firstName" id="firstName" value="">
      </div>

      <div>
        <label for="lastName">Last name</label>
        <input type="text" name="lastName" id="lastName" value="">
      </div>

      <div>
        <label for="password1">Choose a password</label>
        <input type="password" name="password1" id="password1" value="">
      </div>

      <div>
        <label for="password2">Retype password</label>
        <input type="password" name="password2" id="password2" value="">
      </div>

      <div>
        <label for="course">Enrolled in</label>
        <select name="course" id="course" size="1">
          <option value="WebCert4">Web Cert 4</option>
          <option value="WebDiploma">Web Diploma</option>
          <option value="ProgrammingCert4">Programming Cert 4</option>
        </select>
      </div>

      <div>
        <input type="radio" name="mode" id="modeFT" value="FT">
        <label for="modeFT" class="shorterLabel">Full time</label>
        <input type="radio" name="mode" id="modePT" value="PT">
        <label for="modePT" class="shorterLabel">Part time</label>
      </div>
    </fieldset>
  </form>
</body>
</html>
```

```

<div>
  <label for="newsletter" class="longerLabel">
    Do you want to receive our newsletter?</label>
  <input type="checkbox" name="newsletter" id="newsletter" value="yes">
</div>

<div>
  <label for="comments">Any comments?</label>
  <textarea name="comments" id="comments" rows="4" cols="50"> </textarea>
</div>

<div>
  <button type="submit" name="submitButton" id="submitButton">Send Details
</button>
</div>
</fieldset>
</form>

</body>
</html>

```

Have a look at the form in the browser.

Now create the php file that will handle the data. Make sure you name it processRegistration.php, or whatever you supplied for the action attribute.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <title>Process Registration</title>
6 </head>
7
8 <body>
9 <h1>Thank you</h1>
10
11 <p>Thank you for registering. Here is the information you submitted:</p>
12 <dl>
13   <dt>First name</dt>
14   <dd><?= $_POST["firstName"]?></dd>
15
16   <dt>Last name</dt>
17   <dd><?= $_POST["lastName"]?></dd>
18
19   <dt>Password</dt>
20   <dd><?= $_POST["password1"]?></dd>
21
22   <dt>Retyped password</dt>
23   <dd><?= $_POST["password2"]?></dd>
24
25   <dt>Study Mode</dt>
26   <dd><?= $_POST["mode"]?></dd>
27
28   <dt>Course</dt>
29   <dd><?= $_POST["course"]?></dd>
30
31   <dt>Do you want to receive our newsletter?</dt>
32   <dd><?= $_POST["newsletter"]?></dd>

```

```

33
34     <dt>Comments</dt>
35     <dd><?= $_POST["comments"]?></dd>
36 </dl>
37 </body>
38
39 </html>

```

Handling empty form fields

Sometimes users don't enter in all the content when completing a form. This table lists what is sent to the server if no data is entered by the user:

Field	What happens if it's left empty or not selected
Text input field	The field name is sent, along with an empty value.
Password field	The field name is sent, along with an empty value.
Checkbox field	Nothing is sent.
Radio button field	Nothing is sent.
Submit button	Nothing is sent at all if the button isn't clicked.
Reset button	Nothing is ever sent.
File upload	The field name is sent, along with an empty value.
Hidden field	The field name is sent, along with an empty value.
Image field	Same behaviour as a submit button.
Push button	A <button type="button"> does not submit the form so nothing is sent.
Drop down list	Impossible to select no option, so a value is always sent.
Drop down list with multiple attribute	Nothing is sent.
Text area field	The field name is sent, along with an empty value.

When nothing is sent at all for a field, PHP doesn't create an element for the field in the `$_POST`, `$_GET`, or `$_REQUEST` array. So, if you attempt to access the element, you'll generate a PHP notice along the lines of:

PHP Notice: Undefined index: abc in filename.php on line xx

Depending on your server settings notices may not be displayed in the browser. Notices do not stop your code from executing but they should be avoided. This helps to ensure the robustness and security of your application. This means that you should check for the presence of a submitted form field before using it, rather than assuming it exists. You can do this using PHP functions such as `isset()`, `empty()` or `array_key_exists()` :

```

<?php
//check if values have been set
if (isset($_POST["mode"]))
{
    $mode = $_POST["mode"];
}
else
{
    $mode = "Not supplied";
}
?>

```

Or

```

if(array_key_exists("mode", $_POST))
{
    $mode = $_POST["mode"];
}
else
{
    $mode = "Not supplied";
}
Or
if(empty($_POST["mode"]))
{
    $mode = "Not supplied";
}
else
{
    $mode = $_POST["mode"];
}

```

For the variables that are still set but empty you can use empty() or the strlen()

```

if(empty($_POST["firstName"]))
{
    $firstName = "Not supplied";
}
else
{
    $firstName = $_POST["firstName"];
}
Or
if(isset($_POST["firstName"]) && strlen($_POST["firstName"]) > 0)
{
    $firstName = $_POST["firstName"];
}
else
{
    $firstName = "Not supplied";
}
Or
if(isset($_POST["firstName"]) && ($_POST["firstName"]))
{
    $firstName = $_POST["firstName"];
}
else
{
    $firstName = "Not supplied";
}

```

<http://php.net/manual/en/function.array-key-exists.php>

Activity 2. Checking for empty fields

Modify the previous activity so you are checking if mode and newsletter are provided.

```

1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8">
5 <title>Process Registration</title>
6 </head>
7
8 <body>

```

```

9      <?php
10     //check if values have been set using isset and empty
11     if (isset($_POST["newsletter"]))
12     {
13         $newsletter = $_POST["newsletter"];
14     }
15     else
16     {
17         $newsletter = "No";
18     }
19
20     if(!empty($_POST["mode"]))
21     {
22         $mode = $_POST["mode"];
23     }
24     else
25     {
26         $mode = "Not supplied";
27     }
28
29     ?>
30     <h1>Thank you</h1>
31
32     <p>Thank you for registering. Here is the information you submitted:</p>
33     <dl>
34         <dt>First name</dt>
35         <dd><?= $_POST["firstName"] ?></dd>
36
37         <dt>Last name</dt>
38         <dd><?= $_POST["lastName"] ?></dd>
39
40         <dt>Password</dt>
41         <dd><?= $_POST["password1"] ?></dd>
42
43         <dt>Retyped password</dt>
44         <dd><?= $_POST["password2"] ?></dd>
45
46         <dt>Mode</dt>
47         <dd><?= $mode ?></dd>
48
49         <dt>Course</dt>
50         <dd><?= $_POST["course"] ?></dd>
51
52         <dt>Do you want to receive our newsletter?</dt>
53         <dd><?= $newsletter ?></dd>
54
55         <dt>Comments</dt>
56         <dd><?= $_POST["comments"] ?></dd>
57     </dl>
58 </body>
59
60 </html>

```

Additional practice

1. Check all fields are set and are not empty before displaying them.

Dealing with multi value fields

Some form controls such as list boxes and checkboxes allow for more than one value to be selected by the user. To access these values in PHP add square brackets (`[]`) after the field name in your HTML form. When the PHP engine sees a submitted form field name with square brackets at the end, it creates an array of values within the `$_GET` or `$_POST` (and `$_REQUEST`) super global array, rather than a single value. You can access the individual values out of that array.

Activity 3. Dealing with multi value fields

Change the dropdown list to be displayed as a list box and allow multiple selection:

```
<label for="course">Select the course you have studied</label>
<select name="course[]" id="course" size="3" multiple>
  <option value="WebCert4">Web Cert 4</option>
  <option value="WebDiploma">Web Diploma</option>
  <option value="ProgrammingCert4">Programming Cert 4</option>
</select>
```

Change the newsletter checkbox to contain 2 checkboxes:

```
<label for="newsletter">Do you want to receive our newsletter?</label>
<input type="checkbox" name="newsletter[]" id="newsletter" value="news">

<label for="specials">Do you want to receive our specials?</label>
<input type="checkbox" name="newsletter[]" id="specials" value="specials">
```

Now change the php file to deal with the multi fields:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Process Registration</title>
6 </head>
7
8 <body>
9 <?php
10 //check if values have been set
11 if (isset($_POST["mode"]))
12 {
13     $mode = $_POST["mode"];
14 }
15 else
16 {
17     $mode = "Not supplied";
18 }
19
20 if(empty($_POST["firstName"]))
21 {
22     $firstName = "Not supplied";
23 }
24 else
25 {
26     $firstName = $_POST["firstName"];
27 }
28
29 if(isset($_POST["lastName"]) && strlen($_POST["lastName"]) == 0)
30 {
31     $lastName = "Not supplied";
32 }
33 else
34 {
```

```

35     $lastName = $_POST["lastName"];
36 }
37
38
39 if(isset($_POST["password1"]) && (!$_POST["password1"]))
40 {
41     $password1 = "Not supplied";
42 }
43 else
44 {
45     $password1 = $_POST["password1"];
46 }
47
48 if(empty($_POST["password2"]))
49 {
50     $password2 = "Not supplied";
51 }
52 else
53 {
54     $password2 = $_POST["password2"];
55 }
56
57 if(empty($_POST["comments"]))
58 {
59     $comments = "Not supplied";
60 }
61 else
62 {
63     $comments = $_POST["comments"];
64 }
65
66 //process multi selections
67 $coursesStudied = "";
68 $newsletters = "";
69
70 if(isset($_POST["course"]))
71 {
72     foreach ($_POST["course"] as $course)
73     {
74         $coursesStudied = $coursesStudied . $course . ", ";
75     }
76 }
77
78 if(isset($_POST["newsletter"]))
79 {
80     foreach ($_POST["newsletter"] as $newsletter)
81     {
82         $newsletters = $newsletters . $newsletter . ", ";
83     }
84 }
85
86 //remove last comma
87 $coursesStudied = rtrim($coursesStudied, ', ');
88 $newsletters = rtrim($newsletters, ', ');
89 ?>
90 <h1>Thank you</h1>
91
92 <p>Thank you for registering. Here is the information you submitted:</p>
93

```



```

94     <dl>
95         <dt>First name</dt>
96         <dd><?= $firstName ?></dd>
97
98         <dt>Last name</dt>
99         <dd><?= $lastName ?></dd>
100
101         <dt>Password</dt>
102         <dd><?= $password1 ?></dd>
103
104         <dt>Retyped password</dt>
105         <dd><?= $password2 ?></dd>
106
107         <dt>Study mode</dt>
108         <dd><?= $mode ?></dd>
109
110         <dt>Course</dt>
111         <dd><?= $coursesStudied ?></dd>
112
113         <dt>Do you want to receive our newsletter?</dt>
114         <dd><?= $newsletters ?></dd>
115
116         <dt>Comments</dt>
117         <dd><?= $comments ?></dd>
118     </dl>
119 </body>
120
121 </html>

```

rtrim

Notice the use of rtrim to remove the last comma and space:

```

$coursesStudied = rtrim($coursesStudied, ', ');
$newsletters = rtrim($newsletters, ', ');

```

Passing data between servers and browsers

When a browser requests a page to be displayed it is using the HTTP (Hyper Text Transfer Protocol). The protocol specifies the way the page request is sent to the server. The most common request is a GET, what is sent when a browser asks for content. Also common is POST, the request sent when a browser wants to send data to the server.

Stateless

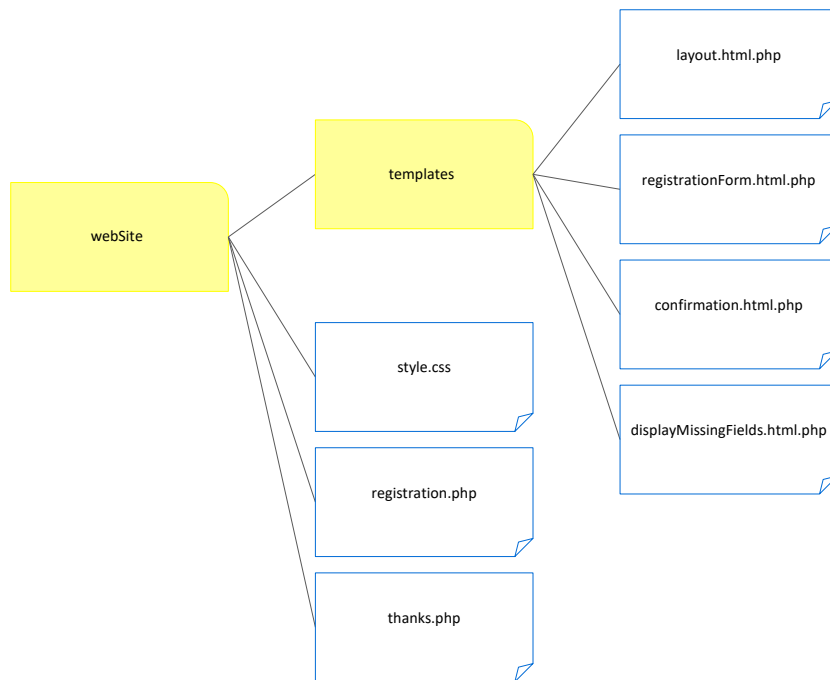
HTTP is called as a stateless protocol because each request is executed independently, without any knowledge of the requests that were executed before it. This means that when we check a form for errors if we want to display to the user a message indicating some data is missing, we need to do some work to redisplay the data they did enter. Since there is no record of previous interactions and data, each interaction request has to be handled based entirely on information that comes with it. The data entered by user or stored in fields.

Error checking web forms

You can use PHP code to validate the web form created. These errors are picked up by the PHP (server-side code). To implement client-side validation, JavaScript code should be used. It is a good idea to validate on both the client and the server.

Activity 4. Keeping values between page requests

If the user is missing some fields, it would be helpful to keep the values they did type in. Download the starting files and make the required changes.



The final code will look like the following files:

Templates

templates/layout.html.php

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title><?= $title ?></title>
  <link href="style.css" rel="stylesheet">
</head>
<body>
  <header>
    <h1>Company Name</h1>
    <nav>
      <ul>
        <li><a href="page1.html">Page 1</a></li>
        <li><a href="page2.html">Page 2</a></li>
        <li><a href="page3.html">Page 3</a></li>
      </ul>
    </nav>
  </header>
  <section>
    <?= $output ?>
  </section>
  <footer>
    <p>&copy; <?= date("Y") ?></p>
  </footer>
</body>
</html>
  
```

<p>Please fill out the details to register</p>

<form action="registration.php" method="post">

<fieldset>

<legend>Details</legend>

<p>

<label for="firstName">First name *</label>

<input type="text" name="firstName" id="firstName" value="<?= setValue("firstName") ?>">

</p>

<p>

<label for="lastName">Last name *</label>

<input type="text" name="lastName" id="lastName" value="<?= setValue("lastName") ?>">

</p>

<p>

<label for="email">Email *</label>

<input type="email" name="email" id="email" value="<?= setValue("email") ?>">

</p>

<p>

<label for="phone">Phone</label>

<input type="text" name="phone" id="phone" value="<?= setValue("phone") ?>">

</p>

<p>

<label>Semester studied *</label>

<label for="semester1" class="radioLabel">Semester 1</label>

<input type="radio" name="semester" id="semester1" value="1" <?= setChecked("semester", "1") ?>>

<label for="semester2" class="radioLabel">Semester 2</label>

<input type="radio" name="semester" id="semester2" value="2" <?= setChecked("semester", "2") ?>>

</p>

<p>

<label for="course">Select the course you have studied</label>

<select name="course" id="course">

<option value="WebCert4" <?= setSelected("course", "WebCert4") ?>>

Web Cert 4</option>

<option value="WebDiploma" <?= setSelected("course", "WebDiploma") ?>>

Web Diploma</option>

<option value="ProgrammingCert4"

<?= setSelected("course", "ProgrammingCert4") ?>>

Programming Cert 4</option>

</select>

</p>

<p>

<label for="newsletter">Do you want to receive our newsletter?</label>

<input type="checkbox" name="newsletter" id="newsletter" value="yes"

```

        <?= isChecked("newsletter", "yes")?>>
    </p>

    <p>
    <label for="comments">Any comments?</label>
    <textarea name="comments" id="comments" rows="4" cols="50">
    <?= setValue("comments") ?></textarea>
    </p>

</fieldset>

    <p>
    <input type="submit" name="submitButton" id="submitButton" value="Send Details">

    </p>
</form>

```

templates/displayMissingFields.html.php

```

<h3>Missing fields</h3>
<p>Please supply the following details:</p>
<ul>
    <?php foreach($missingFields as $field): ?>
        <li><?= $field ?></li>
    <?php endforeach; ?>
</ul>

```

templates/confirmation.html.php

```

<h2>Thank you</h2>
<p>Your registration has been received</p>

```

registration.php

```

<?php
$title = "Registration form";

//start buffer
ob_start();

if(isset($_POST["submitButton"]))
{
    //check for missing fields
    $requiredFields = ["firstName" => "First name", "lastName" => "last name", "email" =>
    "Email", "semester" => "Semester studied"];
    $missingFields = [];

    foreach($requiredFields as $field => $description )
    {
        if (!isset($_POST[$field]) || !$_POST[$field])
        {
            //if the field is missing add it to the $missingFields array
            $missingFields[$field] = $description;
        }
    }

    //display missing fields
    if($missingFields)
    {
        //include missing fields file
        include "templates/displayMissingFields.html.php";
        include "templates/registrationForm.html.php";
    }
}

```

```

    else
    {
        //redirect to thanks.php
        header("Location: thanks.php");
        ob_get_clean();
        exit();
    }
}
else
{
    //first time page is loaded display the form
    include "templates/registrationForm.html.php";
}

function setValue($fieldName)
{
    if(isset($_POST[$fieldName]))
    {
        return htmlspecialchars($_POST[$fieldName]);
    }
}

function setChecked($fieldName, $fieldValue)
{
    if (isset($_POST[$fieldName]) && $_POST[$fieldName] == $fieldValue)
    {
        return "checked";
    }
}

function setSelected($fieldName, $fieldValue)
{
    if (isset($_POST[$fieldName]) && $_POST[$fieldName] == $fieldValue)
    {
        return "selected";
    }
}

$output = ob_get_clean();

include "templates/layout.html.php";

?>

```

```

thanks.php
<?php
$title = "Thank you";

//start buffer
ob_start();

//display thanks file
include "templates/confirmation.html.php";

$output = ob_get_clean();

include "templates/layout.html.php";

?>

```